



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Programmeerimise süvendatud algkursus ITI0140

2013



Teemad

- Järjendi konstrueerimine (ingl *list comprehension*)
- Lambda funktsioon (ingl *lambda function*)
- Generaator (ingl *generator*)



Teemad

- Järjendi konstrueerimine (ingl *list comprehension*)
- Lambda funktsioon (ingl *lambda function*)
- Generaator (ingl *generator*)



Järjendi konstrueerimine

```
squares = []  
for x in range(10):  
    squares.append(x**2)  
print(squares)  
>>> [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```



Järjendi konstrueerimine

```
squares = [x**2 for x in range(10)]
```

```
print(squares)
```

```
>>> [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```



Järjendi konstrueerimine

$[(x, y) \text{ for } x \text{ in } [1, 2, 3] \text{ for } y \text{ in } [3, 1, 4] \text{ if } x \neq y]$

$\gg> [(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)]$



Lambda funktstioon

```
def f (x):  
    return x**2  
print f(8)  
>>> 64
```

```
g = lambda x: x**2  
print g(8)  
>>> 64
```



Lambda funktsoon

```
l = [2, 18, 9, 22, 17, 24, 8, 12, 27]  
print filter(lambda x: x % 3 == 0, l)  
>>> [18, 9, 24, 12, 27]
```




Lambda funktsooon

```
l = [2, 18, 9, 22, 17, 24, 8, 12, 27]  
print map(lambda x: x * 2 + 10, l)  
>>> [14, 46, 28, 54, 44, 58, 26, 34, 64]
```



Generaator

```
def gen():  
    yield 3  
    yield 5  
    yield 9
```

```
def main():  
    x = gen()  
    for i in range(4):  
        print(next(x))
```

```
>>> 3
```

```
>>> 5
```

```
>>> 9
```

```
Traceback (most recent call last):
```

```
    print(next(x))
```

```
StopIteration
```



Generaator

```
def reverse(data):  
    for index in range(len(data)-1, -1, -1):  
        yield data[index]
```

```
for char in reverse('golf'):  
    print(char)
```

```
>>> f  
l  
o  
g
```



Generaator

```
def counter (maximum):  
    i = 0  
    while i < maximum:  
        yield i  
        i += 1
```

```
x = counter(4)  
    for i in x:  
        print(i)
```

```
>>> 0
```

```
1
```

```
2
```

```
3
```



Generaator

```
def counter (maximum):  
    i = 0  
    while i < maximum:  
        yield i  
        i += 1
```

```
x = list(counter(4))  
print(x)  
>>> [0, 1, 2, 3]
```



Ülesanne

Kirjutada generaatorfunktsioon `carcosa.txt` faili peale, mis tagastaks uue sõna iga järgneva väljakutse peale. Kasutades tehtud generaatorfunktsiooni, genereerida järjend (list comprehensioni abil), mis sisaldab igat kolmandat sõna, kokku 100 tükki. Selle peale kirjutada lambda-funktsiooniga filter, mis väljastaks sellest listist sõnad, mille pikkus on suurem kui andtud listi keskmine sõnapikkus. Väljastada sõnad suure algustähega.